Evaluating Variance and Reliability of LLM Powered Semantic Operators Under Ambiguous Queries Using LOTUS

Geetansh Kumar

***Abstract*—Large Language Models (LLMs) are increasingly being used as semantic operators for executing natural language queries over unstructured data. Frameworks such as LOTUS enable users to express complex analytical tasks in natural language, allowing LLMs to interpret and execute semantic operations directly over datasets and documents. While this provides flexibility and accessibility beyond traditional query systems, it also introduces challenges related to ambiguity, reasoning consistency, and non-deterministic behavior. In this work, we evaluate the reliability and variance of LLM-powered semantic operators implemented through LOTUS under ambiguous and unambiguous query settings. We construct a SQL-based gold standard dataset by combining the latest IMDb dataset with the research-backed JOB benchmark using fuzzy matching techniques. In addition, we create an equivalent document-based dataset by scraping and processing IMDb HTML pages to simulate unstructured retrieval and reasoning tasks.**

Our experiments analyze output variance across multiple runs, accuracy relative to structured SQL ground truth, and differences in reasoning behavior between ambiguous and unambiguous queries. We further investigate practical system limitations such as context-window constraints, API rate limits, and LOTUS scalability challenges for large documents. Our findings show that ambiguity significantly increases output variance and decreases reliability, while large document settings introduce additional instability due to retrieval and reasoning complexity.

I. Introduction

RECENT advances in Large Language Models (LLMs) have enabled a new paradigm of querying and reasoning over data using natural language. Instead of relying exclusively on traditional SQL queries or symbolic database operators, frameworks such as LOTUS allow users to express analytical intent directly in natural language. The framework interprets these requests and executes semantic operations over unstructured document collections.

Unlike traditional query systems, LLM-powered semantic operators can capture intent even when queries are loosely specified or conversational in nature. This flexibility enables more intuitive interactions with data and reduces the need for domain-specific query languages. However, this same flexibility introduces challenges related to ambiguity and probabilistic reasoning.

Ambiguous natural language queries may lead to multiple valid interpretations, causing variability in outputs across executions. Even when queries are unambiguous, LLMs may still exhibit non-deterministic behavior due to probabilistic token generation and contextual reasoning differences. These properties make reliability and reproducibility important concerns for semantic query systems.

This work focuses on evaluating how ambiguity affects the behavior of semantic operators implemented using LOTUS. Specifically, we analyze:

- Variance in outputs across multiple runs
- Accuracy relative to structured SQL-based ground truth
- Differences in reasoning behavior between ambiguous and unambiguous queries
- Practical limitations involving context windows and scalability

Our goal is to better understand the reliability and usefulness of LLM-powered semantic systems for data querying and analysis task

II. Background and Motivation

Traditional database systems rely on deterministic query execution mechanisms where the same query over the same data produces identical results. In contrast, LLM-powered systems introduce semantic reasoning into the execution pipeline. Instead of exact symbolic matching, these systems interpret meaning, context, and intent.

LOTUS represents one such framework that formalizes semantic operators powered by LLMs. These operators can perform tasks such as:

- Semantic filtering
- Reasoning-based selection
- Natural language aggregation
- Document interpretation

While these capabilities are powerful, they create challenges that do not exist in traditional query systems:

- Ambiguous language may produce inconsistent interpretations.
- Retrieval over unstructured documents can introduce noise.
- Large document contexts may exceed practical context-window limits.
- Repeated executions can yield different outputs.
- Rate limits and API throughput constraints complicate large-scale experimentation.

Our work investigates these issues systematically by comparing LOTUS outputs against a structured SQL based gold standard

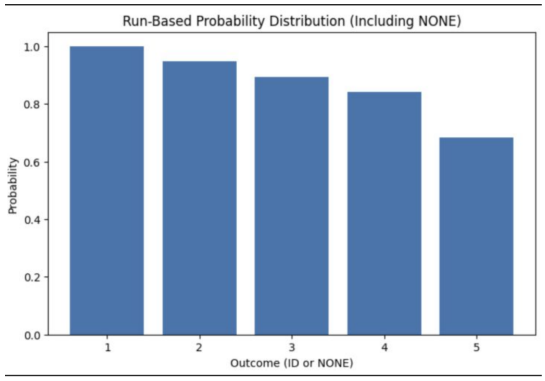

Fig. 1. Variance

III. Dataset Construction

A. Structured Dataset

To construct a reliable evaluation benchmark, we combine information from two primary sources: The latest IMDb dataset and The research-backed JOB benchmark dataset.

The IMDb dataset provides extensive metadata related to movies, cast information, production companies, ratings, and associated entities. The JOB benchmark contributes a collection of structured relational queries widely used in database research.

We perform fuzzy matching between the datasets using movie title similarity and release year alignment to integrate related entities and enrich the overall information available for evaluation. Entries that do not satisfy both conditions are discarded during integration. This process enables us to combine the breadth of IMDb metadata with the structured query reliability of the JOB benchmark. The resulting relational dataset serves as our SQL-based ground truth. Queries from the JOB benchmark are executed against the JOB schema enriched with IMDb metadata to generate deterministic baseline outputs.

```sql
SELECT DISTINCT t.*
FROM temp_tables.small_title t
JOIN temp_tables.small_title_ratings tr
    ON t.tconst = tr.tconst
JOIN temp_tables.title_mapping m
    ON t.tconst = m.tconst
JOIN temp_tables.small_movie_companies mc
    ON m.job_movie_id = mc.movie_id
JOIN temp_tables.small_company_name cn
    ON mc.company_id = cn.id
WHERE cn.country_code = '[us]'
  AND (t.genres LIKE '%Documentary%' OR
       t.genres LIKE '%Horror%')
  AND tr.averageRating > 5.0
  AND CAST(t.startYear AS INT) BETWEEN 2000 AND 2020;
```

tconst	titleType	primaryTitle	originalTitle
tt1363109	tvMovie	Journey to the Edge of the Universe	Journey to the Edge of the Universe
tt0810412	short	The Legend of Flashpants	The Legend of Flashpants
tt0497557	short	Don't Whistle	Don't Whistle
tt0443488	short	Dream on Silly Dreamer	Dream on Silly Dreamer

Fig. 2. SQL Query

B. Fuzzy Matching Strategy

Exact entity matching between large datasets is often impractical due to formatting inconsistencies, naming variations, abbreviations, and missing fields. To address this, we use fuzzy matching techniques during dataset integration.

The fuzzy matching process is applied only during dataset integration. Matching is performed using movie title similarity together with release year alignment. Records failing these matching conditions are discarded to maintain dataset consistency. This integration strategy increases dataset coverage while preserving alignment quality between the IMDb and JOB datasets.

C. Unstructured dataset

To evaluate LOTUS in document-oriented environments, we convert the structured relational data into an unstructured HTML-based corpus.

We scrape IMDb pages corresponding to movie titles, cast information, production companies, ratings, and related metadata. The scraped HTML is filtered to retain only semantically relevant information before being used for LOTUS-based querying. The resulting documents simulate realistic web-based information retrieval environments.

D. HTML Processing and Optimization

Raw HTML contains significant amounts of irrelevant information that increases context size and retrieval cost. Therefore, we preprocess and filter the HTML content before experimentation. Our preprocessing pipeline includes:

- Removing unnecessary scripts and styling
- Extracting semantically relevant sections
- Reducing redundant metadata
- Structuring documents for efficient retrieval
- Rate limits and API throughput constraints complicate large-scale experimentation.

This optimization step is critical because large documents substantially increase LLM context usage, latency, and API cost. We ensure that the processed HTML corpus retains the same semantic information present in the relational dataset to maintain evaluation consistency.

IV. Experimentation

A. Framework

All experiments are conducted using the LOTUS framework. LOTUS enables semantic operators that execute natural language reasoning over unstructured document-based data. In our experiments, LOTUS operates primarily over the document-based HTML dataset while SQL execution over the structured dataset provides the reference ground truth.

```
PROMPT_UNIFIED_GENRE_12A = (
    """
{text_cleaned} is about a title made by a U.S. company,
and title should capture real-life events or feels like the kind of movie that keeps me up at night.
It should have an average rating above 5.
It should also have been released between 2000 through 2020.
"""
)
```

	id	text_cleaned	explanation_filter	raw_output_filter
0	tt0427969	Cast & crew\nUser reviews\nTrivia\nFAQ\nIMDbPr...	- Production: The page lists U.S. production(d...	Reasoning:\n- Production: The page lists U.S. ...
1	tt0443488	Cast & crew\nUser reviews\nIMDbPro\nAll topics...	- The page describes "Dream on Silly Dreamer" ...	Reasoning:\n- The page describes "Dream on Sil...
2	tt0482572	Cast & crew\nUser reviews\nTrivia\nFAQ\nIMDbPr...	- The page is for Pride and Glory (title shown...	Reasoning:\n- The page is for Pride and Glory ...
3	tt0497557	Cast & crew\nIMDbPro\nAll topics\nDon't Whistl...	- The page is for the title "Don't Whistle" (r...	Reasoning:\n- The page is for the title "Don't...
4	tt0810412	Cast & crew\nIMDbPro\nAll topics\nThe Legend o...	- The page lists Country of origin: United Sta...	Reasoning:\n- The page lists Country of origin...
5	tt1363109	Cast & crew\nUser reviews\nTrivia\nIMDbPro\nAl...	- Title: "Journey to the Edge of the Universe"...	Reasoning:\n- Title: "Journey to the Edge of t...

Fig. 3. Semantic Query

B. Evaluation Metrics

Accuracy
We compare LOTUS outputs against the SQL-based gold standard derived from JOB benchmark queries. Because semantic outputs may differ slightly in formatting or wording, we use fuzzy matching to align outputs with ground truth.
Variance To measure output consistency, each query is executed multiple times. We analyze:

- Output stability across runs
- Variability in retrieved entities
- Differences in generated reasoning paths

Reasoning Behavior We compare how LOTUS interprets ambiguous versus unambiguous queries. This includes manual inspection of the reasoning traces and semantic explanations produced by LOTUS during execution, allowing us to analyze how interpretation changes between ambiguous and unambiguous queries.

C. Context Window and Rate Limit Challenges

During experimentation, we observe several practical limitations involving LOTUS and LLM inference. Context Window Constraints Large HTML documents significantly increase prompt size and retrieval overhead. As document size grows, LOTUS struggles to efficiently process the full context. This leads to:

- Increased latency
- Context truncation risks
- Reduced reasoning consistency
- Higher API cost

API Rate Limits Experimentation using multiple LLMs generates substantial inference load and exposes practical API throughput limitations. Rate limiting introduces additional execution constraints, particularly when processing large batches of queries.

D. Batching and Chunking Strategy

To address these issues, we implement custom batching and chunking strategies. Our approach includes:

- Splitting large documents into smaller semantic chunks
- Reducing prompt size per request
- Executing incrementally
- Grouping queries into controlled execution batches

These optimizations improve throughput, reduce context overload, lower rate-limit failures, and stabilize LOTUS execution behavior during large-scale evaluation.

V. Results

A. Impact of Ambiguity

Our experiments show that ambiguous queries produce significantly higher variance than unambiguous queries.

This demonstrates that LOTUS semantic operators are sensitive to ambiguity.

B. Accuracy Trends

Unambiguous queries generally achieve higher accuracy relative to the SQL-based ground truth. However, ambiguous queries show reduced accuracy because LOTUS may select interpretations that differ from the intended semantics.

C. False Positive and Overestimation Behavior

We observe a strong tendency toward false positives and overestimation in semantic outputs. Because fuzzy matching is used only during dataset integration and semantic retrieval favors broader matches, predicted outputs are frequently equal to or larger than the SQL ground truth. Underestimation occurs much less frequently, producing an asymmetric error profile.

D. Non-Distributional Output Behavior

The outputs generated by LOTUS cannot be cleanly modeled as a standard probability distribution. Instead of centering around a true answer with symmetric variance, outputs tend to concentrate around a subset of dominant interpretations and semantically broader matches. This means traditional statistical assumptions such as normality do not hold well for semantic operator evaluation.

E. One-Sided Error Characteristics

Given the large number of false positives and the overestimation tendency, the evaluation problem behaves more like a one-sided error estimation task. Rather than modeling symmetric deviation around ground truth, we focus on:

- Run-to-run variation
- Deviation from SQL outputs
- Stability across executions

This provides a more meaningful representation of semantic operator reliability

F. Large Document Limitations

We observe that LOTUS is not highly optimized for very large documents. Performance degrades as:

- Context size increases
- Retrieval complexity grows
- More document chunks are required

These limitations lead to:

- Increased execution latency
- Higher inconsistency
- Reduced retrieval precision

G. Key Findings

Our overall findings indicate:

- LOTUS performs reliably for well-defined queries.
- Ambiguity significantly increases variance and inconsistency.
- Semantic outputs exhibit non-standard statistical behavior.
- Practical deployment requires careful batching and chunking strategies.

VI. Discussion and Conclusion

The results demonstrate that LLM-powered semantic operators behave fundamentally differently from traditional deterministic database systems.

While LOTUS enables highly flexible natural language querying over unstructured data, ambiguity propagates directly into semantic execution behavior. This creates challenges for reproducibility, evaluation, and trustworthiness. Our experiments show that ambiguity significantly impacts consistency and accuracy, while large document settings introduce additional retrieval and context-related instability. We also observe that semantic outputs do not behave like a standard probabilistic distribution centered around the true answer. Instead, outputs are concentrated around dominant interpretations and broader semantic matches. A major observation from our evaluation is the prevalence of false positives and overestimation behavior. Since underestimation occurs less frequently, the resulting error profile behaves more like a one-sided error estimation problem rather than symmetric statistical variance.

The study also highlights important practical limitations of LOTUS and LLM-based querying systems, including:

- Context-window constraints for large documents
- Rate-limit and throughput challenges
- Increased latency and inference cost

To address these issues, batching and chunking strategies become necessary for scalable experimentation and execution. Overall, LOTUS demonstrates strong potential for flexible semantic querying, particularly for well-defined natural language queries.

These findings provide insight into the limitations and future directions of LLM-powered semantic operator systems.

References

1) LOTUS Framework Documentation: https://lotus-data.github.io/
2) IMDb Dataset: https://datasets.imdbws.com/
3) Join Order Benchmark (JOB): https://github.com/gregrahn/join-order-benchmark
4) DocETL: A Framework for LLM-Powered Document Processing Pipelines. https://github.com/ucbepic/docetl
5) “Non-determinism of ‘deterministic’ LLM settings.” arXiv preprint arXiv:2408.04667v5, 2025.